Journal of Computational and Applied Mathematics 22 (1988) 257–270
North-Holland

Recent developments in constrained optimization

Philip E. GILL, Walter MURRAY, Michael A. SAUNDERS and Margaret H. WRIGHT
Systems Optimization Laboratory, Department of Operations Research, Stanford University, Stanford, CA 94305, U.S.A.

Received 1 April 1987

Keywords: Optimization, nonlinear programming, nonlinear constraints.

1. Introduction

Constrained optimization problems occur in many applications of science, engineering, social science and medicine, and hence it is useful for practitioners in these areas to be informed about numerical methods for solving such problems. However, improvements in optimization typically do not spread quickly to the non-specialist literature. This paper contains a brief summary of two recent developments in the solution of nonlinearly constrained problems: direct use of the Lagrangian function, and linearization of nonlinear constraints. In order to highlight the new approaches, we also discuss techniques used in older methods. To illustrate the capabilities of modern optimization methods, a brief description is given of the development of methods to solve large nonlinearly constrained problems that arise in the electrical power industry.

The general nonlinear programming problem may be stated in the following form:

$$\underset{x \in \mathbb{R}^n}{\text{minimize}}\, F(x) \quad \text{subject to } l \leqslant \begin{pmatrix} x \\ A_L x \\ c(x) \end{pmatrix} \leqslant u;$$

where $F(x)$ is a smooth nonlinear objective function, A_L is a constant matrix of constraints, and $c(x)$ is a vector of smooth nonlinear constraint functions. Note that simple bounds and linear constraints are represented separately from nonlinear constraints; this distinction is usually made during the solution process because of the increased efficiency resulting from separate treatment of linear constraints. This article will concentrate almost entirely on the treatment of nonlinear constraints because procedures for bound and linear constraints are relatively well understood. (For a more detailed discussion, see, e.g., [18] and [24].) Hence, we shall consider primarily the following problem:

$$\text{NP:} \quad \underset{x \in \mathbb{R}^n}{\text{minimize}}\, F(x) \quad \text{subject to } c(x) \geqslant 0,$$

where F and c have continuous second derivatives, at least in the neighborhood of the solution. (Such smoothness is crucial to the methods discussed in this article.) A solution of NP will be

denoted by x^*. Let $g(x)$ denote the gradient of F, m the number of constraints (the dimension of c), $\mathscr{A}(x)$ the $m * n$ Jacobian matrix of c, and $A(x)$ the Jacobian matrix of the subset of constraints active at x^*. A point x such that $c(x) \geqslant 0$ is said to be *feasible*; the constraint c_i is said to be *violated* at x if $c_i(x) < 0$. The solution of NP is unique under certain conditions, but it is often difficult to determine whether the conditions are satisfied. Consequently, we shall be concerned only with determining *local minima* of NP. (Methods for global optimization usually involve repeated use of methods for finding local minima.)

2. Complications of constraint nonlinearities

It is widely agreed that problems with nonlinear constraints are considerably more difficult than those with purely linear constraints. Broadly speaking, the difficulty arises because of the conflict between reducing F and satisfying the constraints. In an unconstrained problem, the entire aim of a method is to reduce F. This leads to the well known class of *descent methods* (see [35]), in which the $(k+1)$th iterate is defined by

$$x_{k+1} = x_k + \alpha_k p_k, \tag{1}$$

where p_k is an n-dimensional search direction and α_k is a nonnegative *step-length*. If the gradient at x_k is nonzero, p_k and α_k can always be chosen so that $F(x_{k+1})$ is 'sufficiently less' than $F(x_k)$.

Bound and/or general linear constraints add few conceptual difficulties to the basic definition (1) of an iteration. Under mild assumptions, a finite procedure exists for finding a feasible point (or determining that none exists) (see, e.g., [13]). Once a feasible point has been found, all subsequent iterates will remain feasible if p_k and α_k in (1) are chosen appropriately. The additional complexity of dealing with bound and/or general linear constraints is in organization of the linear algebraic operations associated with each iteration—particularly for large problems, where such operations constitute a significant proportion of the total effort required.

In contrast, attaining and maintaining feasibility with respect to nonlinear constraints are potentially infinite (if not impossible) tasks. There is no guaranteed procedure for determining whether a feasible point exists with respect to a set of general nonlinear constraints; furthermore, even if a feasible point is given, moving to another feasible point generally requires an iterative procedure. Thus, any method for NP must somehow balance changes in F against changes in the constraint violations.

The general approach in all methods to be described is to transform problem NP into a sequence of solveable subproblems (or possibly just a single subproblem). The amount of work involved in the subproblems varies considerably among methods. Given the relative ease with which unconstrained problems can be solved (and the availability of software for such problems), most early methods attempted to transform NP into an unconstrained subproblem.

3. Penalty-function methods

A *penalty function* is a combination of the original objective function and a *penalty term* that increases monotonically with some measure of the constraint violations. (For a general treatment of penalty functions, see [17,19].)

The most widely used differentiable penalty term is the sum of squared constraint violations, which gives the *quadratic penalty function*:

$$P_q(x, \rho) = F(x) + \tfrac{1}{2}\rho \sum_{i=1}^{m} [\min(0, c_i(x))]^2. \tag{2}$$

(The quadratic penalty function is usually attributed to Courant; it appears in many guises in different application areas.)

Let $x^*(\rho)$ denote the *unconstrained minimum* of $P_q(x, \rho)$. Under mild conditions, it can be shown that $x^*(\rho)$ approaches x^* as ρ approaches infinity. Thus, the classical quadratic penalty function approach is to find the unconstrained minimum of $P_q(x, \rho)$ for a sequence of increasing values of ρ. Methods of this type reached a height of popularity in the late 1960s, and were often called 'sequential unconstrained minimization techniques' (SUMT).

Since each iteration involves an unconstrained minimization of a differentiable function, the subproblem can be solved using standard methods. Unfortunately, quadratic penalty function methods suffer from certain difficulties. In particular, the penalty parameter ρ in (2) must approach infinity in order for $x^*(\rho)$ to converge to x^*. Each increase in ρ leads to increased ill-conditioning in the unconstrained subproblems, which consequently become more and more difficult for "off-the-shelf' unconstrained routines. Furthermore, these methods are inefficient if each unconstrained subproblem is solved accurately.

These difficulties can be avoided by using a nondifferentiable penalty function, of which the most popular is the l_1 penalty function:

$$P_1(x, \rho) = F(x) + \rho \sum_{i=1}^{m} |\min(0, c_i(x))|^2. \tag{3}$$

Under mild conditions, there is a finite threshold value $\bar{\rho}$ such that x^* is a local minimum of P_1 for $\rho > \bar{\rho}$ [36]; this implies that x^* can be computed with a single unconstrained minimization. (For this reason, penalty functions like P_1 are sometimes termed *exact penalty functions*.)

Note that P_1 is not differentiable at points where a constraint function vanishes — in particular, at x^*. Therefore, special methods must be used to perform the unconstrained minimization (see, e.g., [11], [28] and [8]). Many of these special methods are closely related to the sequential quadratic programming methods discussed in Section 8 (see, e.g., [29] and [9]), and thus will not be considered further.

4. Barrier-function methods

An alternative approach to dealing with inequality constraints is based on the use of barrier functions, which combine the original objective function with a weighted barrier term (a function with a positive singularity at the boundary of the feasible region). The most popular are the logarithmic [20] and inverse [6] barrier functions:

$$B_L(x, \mu) = F(x) - \mu \sum_{i=1}^{m} \ln c_i(x), \qquad B_I(x, \mu) = F(x) + \mu \sum_{i=1}^{m} 1/c_i(x).$$

Let $x^*(\mu)$ denote the *unconstrained minimum* of $B(x, \mu)$. Under mild conditions, it can be shown that $x^*(\mu)$ approaches x^* as μ approaches zero. If a strictly feasible starting point is

known, a barrier-function method will produce a sequence of improving (and strictly interior) estimates of x^*. However, the extreme nonlinearity and special properties of barrier functions make it difficult to apply standard unconstrained methods.

The logarithmic barrier function has recently received attention because of its application in solving linear programs (see, e.g., [23]).

5. Generalized reduced-gradient methods

Another approach to treating nonlinear constraints is a direct analogue of feasible-point methods for linearly constrained problems (see Section 2). The idea is to reduce F while remaining 'on' the nonlinear constraints. Methods of this type were proposed by Rosen [43] and Abadie and Carpentier [2]. The best known such methods are called *generalized reduced-gradient* (GRG) methods. The subproblem in a GRG method usually involves changing x in order to reduce F, followed by an iterative procedure to restore feasibility.

GRG methods have been implemented in several popular software packages, and have been effective on many practical problems. Unfortunately, they can be extremely inefficient when applied to problems with highly nonlinear constraints. GRG methods have tended to be most successful when applied to constraints that are 'almost' linear. Some variations have been developed in which substantial constraint violations are tolerated, but these no longer fit within the original motivation.

6. Properties of the Lagrangian function

The first development in the treatment of nonlinear constraints to be emphasized in this paper is direct use of the Lagrangian function. (Penalty-function and GRG methods use the Lagrangian function only implicitly.) In order to explain methods based on the Lagrangian function, we summarize its role in the optimality conditions for NP. (For a detailed discussion of optimality conditions, see [38].)

At any given point, a certain subset of the constraints (the *active* constraints) will be satisfied exactly. Let c denote the set of constraints active at x^*, and $A(x)$ denote the Jacobian of $c(x)$. Subject to certain conditions on the active constraints, a first-order necessary condition for optimality at x^* is

$$g(x^*) = A(x^*)^{\mathrm{T}}\lambda^* \quad \text{with } \lambda^* \geqslant 0, \tag{4}$$

i.e., the gradient of the objective function must be a non-negative linear combination of the gradients of the active constraints. The relationship (4) implies that x^* is a stationary point (with respect to x) of the Lagrangian function

$$L(x, \lambda) \equiv F(x) - \lambda^{\mathrm{T}} c(x) \tag{5}$$

when $\lambda = \lambda^*$. Let $W(x, \lambda)$ denote the Hessian of the Lagrangian function:

$$W(x, \lambda) \equiv \nabla^2 F(x) - \sum_i \lambda_i \nabla^2 c_i(x).$$

The second-order conditions for optimality involve the behavior of the Lagrangian function within a subspace defined by $A(x^*)$. Let $Z(x)$ denote a basis for the null space of $A(x)$, i.e., the set of vectors orthogonal to the rows of $A(x)$. By definition,

$$A(x)Z(x) = 0.$$

A second-order sufficient condition for x^* to solve NP is that the matrix $Z(x^*)^T W(x^*, \lambda^*)Z(x^*)$ (the projected Hessian of the Lagrangian function) be positive definite; this implies that x^* is a minimum of $L(x, \lambda^*)$ in the subspace defined by $Z(x^*)$.

7. Augmented Lagrangian methods

Augmented Lagrangian methods, which were first suggested by Hestenes [27] and Powell [36], are motivated by properties of the Lagrangian function. The aim is to find a differentiable function of which x^* is an unconstrained minimum, without the ill-conditioning associated with the quadratic penalty function. (The Lagrangian function itself is unsuitable, since x^* is in general only a saddle point of $L(x, \lambda^*)$, and no general reliable methods are known for finding a saddle point.)

Since the Hessian of the Lagrangian function is positive definite at x^* when restricted to the null space of $A(x^*)$, $W(x^*, \lambda^*)$ can display negative curvature only along directions in the range space of $A(x^*)^T$. This suggests 'augmenting' the Lagrangian function (5) by a term that retains the stationary property of x^*, but adds positive curvature in the range space of $A(x^*)^T$. The most popular choice for the additional term is simply the quadratic penalty term from Section 3. (Note that both the quadratic penalty term and its gradient vanish at x^*.) It can be shown that there is a finite threshold $\bar{\rho}$ such that x^* is an unconstrained minimum of

$$F(x) - \lambda^{*T} c(x) + \tfrac{1}{2}\rho c(x)^T c(x) \tag{6}$$

for any $\rho > \bar{\rho}$.

Since the optimal Lagrange multipliers are unknown, an augmented Lagrangian function will be of the form (6), with a Lagrange multiplier estimate replacing λ^*. Let λ_k denote the estimate at x_k; to be acceptable, λ_k must converge to λ^* as x_k approaches x^*. The general structure of an augmented Lagrangian method is the following. Given λ_k and ρ, the new iterate x_{k+1} is the unconstrained minimum of the augmented Lagrangian function

$$L_A(x, \lambda_k, \rho) \equiv F(x) - \lambda_k^T c(x) + \tfrac{1}{2}\rho c(x)^T c(x). \tag{7}$$

A new multiplier estimate λ_{k+1} is then computed at x_{k+1}, and ρ is increased if the constraint violations at x_{k+1} have not decreased sufficiently from those at x_k.

An important feature of augmented Lagrangian methods is that the rate of convergence of $\{x_k\}$ to x^* cannot exceed that of $\{\lambda_k\}$ to λ^*. With certain early suggestions for λ_{k+1}, the multipliers converge only at a *linear rate*. Thus, augmented Lagrangian methods should use multiplier estimates of the highest possible order.

Many implementations of augmented Lagrangian methods have been produced, and they tend to be more efficient and reliable than penalty-function or GRG methods. However, a significant difficulty in practice is the choice of the penalty parameter ρ. (The knowledge that ρ is finite does not imply how to find a good value.) If ρ is too small, L_A may not have an unconstrained

minimum in the neighborhood of x^* (or may be unbounded below). If ρ is too large, the unconstrained subproblems will be difficult to solve because of the ill-conditioning associated with penalty functions. Another drawback of augmented Lagrangian methods is the inefficiency noted above for penalty function methods, i.e., the wasted effort to solve early subproblems accurately. Some authors have suggested techniques for systematically controlling the accuracy to which each subproblem is solved (e.g., [48]).

8. Projected Lagrangian methods

In considering augmented Lagrangian methods, it might seem undesirable to add a penalty term simply to convert x^* from a saddle point to an unconstrained minimum, especially when the choice of a suitable penalty parameter is complicated. Given the optimality properties of x^* in the null space of $A(x^*)$, it is possible to pose instead a linearly constrained subproblem with an objective function that approximates the Lagrangian function. The second major advance in the treatment of nonlinear constraints to be emphasized is the inclusion of constraint linearizations in the subproblem. This development has been possible because good methods are now available for solving the resulting linearly constrained subproblems.

The second-order sufficiency conditions for optimality imply that x^* must be a minimum (with respect to x) of $L(x, \lambda^*)$ when x is restricted to lie in the linear subspace $A(x^*)(x - x^*) = 0$, i.e., x^* is a solution of the minimization problem

$$\underset{x \in \mathbb{R}^n}{\text{minimize}} L(x, \lambda^*) \quad \text{subject to } A(x^*)(x - x^*) = 0. \tag{8}$$

The class of *projected Lagrangian methods* is based on computing the next iterate by solving a subproblem that approximates the 'ideal' problem (8). Let x_k be an estimate of x^*. A 'natural' set of linear constraints is suggested by the usual Taylor-series linearization of c about x_k:

$$c(x_k + p) \approx c_k + A_k p, \tag{9}$$

where c_k and A_k denote $c(x_k)$ and $A(x_k)$. Since $c(x^*) = 0$, we could impose the requirement that the linearized constraints (9) will be satisfied exactly at $x_k + p_k$, i.e., p_k must satisfy the linear constraints

$$A_k p = -c_k. \tag{10}$$

If $x_k = x^*$, the constraints (10) define the same subspace as the constraints in the 'ideal' problem (8). However, a difficulty with the formulation (10) is that in general the correct active set is not known in advance. Therefore, the most popular formulation of the linear constraints of the subproblem includes linearizations of all the nonlinear constraints, which are posed in the subproblem as a set of linear inequality constraints, i.e.

$$\mathscr{A}_k p \geqslant -c_k. \tag{11}$$

It can be shown that, in a neighborhood of x^*, with a suitable formulation of the objective function, the solution of a linearly constrained subproblem with constraints (11) will produce a correct prediction of the active set (i.e., the linearized constraints active in the subproblem are the active nonlinear constraints). We assume henceforth that the constraints of the subproblem are given by (11). (Other possible formulations of the constraints are discussed in [30].)

The derivation of (8) indicates that the objective function of any linearly constrained subproblem should be an approximation to the Lagrangian function. Since the optimal multiplier λ^* is unknown, some procedure is necessary to obtain Lagrange multiplier estimates. The usual strategy is to construct the objective function so that the Lagrange multipliers of the subproblem approach the optimal multipliers as x_k converges to x^*.

An obvious candidate for the objective function of the linearly constrained subproblem is $F(x) - \lambda_k^T c(x)$, a general approximation to the Lagrangian function of (8), using the latest multiplier estimate, However, if $x_k = x^*$ and $\lambda_k = \lambda^*$, the Lagrange multiplier of the resulting subproblem (with constraints (11)) would be zero rather than λ^*. Therefore, the objective function of the subproblem is usually defined instead as

$$F(x) - \lambda_k^T c(x) + \lambda_k^T \mathscr{A}_k x. \tag{12}$$

The additional linear term is (12) alters the gradient so that the multipliers of the subproblem converge to λ^*, but does not change the optimality properties of x^*. Methods in which the subproblem has constraints (11) and objective function (12) were suggested originally by Robinson [42] and Rosen and Kreuser [44]. Their widest application has been in the area of large-scale nonlinear programming (see Section 10 for further discussion).

Alternatively, a quadratic model of the Lagrangian function may be used as the objective function of the subproblem. In this case, the subproblem becomes a quadratic program. Methods of this type have been the subject of intense in recent years because of their remarkable success in practice. They are most commonly called sequential quadratic programming (SQP) methods, and will be considered in some detail in the next two sections.

9. Sequential quadratic programming methods

SQP methods are widely regarded today as the most effective general methods for problem NP. The first SQP method was suggested by Wilson [50]; they were popularized mainly by Biggs [4], Han [25] and Powell [39]. (For a brief history of SQP metods and an extensive bibliography, see [24]. For a survey of recent results and references, see [40].)

In an SQP method, an iteration is defined by (1), and the search direction p_k solves the following subproblem:

$$\underset{p \in \mathbb{R}^n}{\text{minimize}}\, g_k^T p + \tfrac{1}{2} p^T H_k p \quad \text{subject to } \mathscr{A}_k p \geqslant -c_k, \tag{13a,b}$$

where H_k is an approximation to the Hessian of the Lagrangian function. (The form of the objective function (13a) allows the multipliers of the QP subproblem to be taken as an estimate of λ^*.) The steplength α_k in (1) is chosen to produce a sufficient decrease in some suitable merit function, in order to prevent divergence when x_k is far from the solution, or when H_k is a poor approximation to the true Hessian. We now mention two important aspects of an SQP method: the definition of H_k and the selection of a merit function. Most attention to date has focused on defining H_k as a positive-definite quasi-Newton approximation to the Hessian of the Lagrangian function. The best known quasi-Newton update is the BFGS update. Let H_{k-1} denote the previous approximation to the Hessian; H_0 is usually taken as the identity matrix. The updated matrix H_k has the form $H_k = H_{k-1} + U_{k-1}$, where U_{k-1} is a rank-two matrix. An important

property of the BFGS update in the unconstrained case is hereditary positive-definiteness with a suitable choice of step length. If H_{k-1} is positive definite and the step length satisfies certain 'sufficient decrease' criteria, then H_k will also be positive definite. A further desirable feature of the BFGS update is that the iterates converge superlinearly. (For a detailed discussion of quasi-Newton methods, see [14] and [15].)

For the nonlinear-constraint case, we seek a quasi-Newton approximation to the Hessian of the Lagrangian function. However, since x^* is not an unconstrained minimum of the Lagrangian function, it may be impossible, with any line search, to find a step length for which the updated matrix H_k remains positive definite. Using an indefinite matrix H_k in (13a) could lead to difficulties of unboundedness or nonuniqueness of the subproblem solution. Skipping the update when positive-definiteness cannot be retained would destroy the favorable local convergence properties of the BFGS update.

Several strategies have been suggested to overcome these difficulties. Powell [39] defines a modification of the BFGS update for which positive-definiteness is retained and two-step superlinear convergence can be achieved. A second alternative is to consider H_k as an approximation to the Hessian of an augmented Lagrangian function (see, e.g., [26], [48] and [3]), where the penalty parameter must be large enough so that the Hessian remains positive definite. Finally, a perturbation can be added in the range space of the constraint gradients when necessary to retain positive-definiteness (see [22]).

As mentioned in Section 2, the necessity to balance the constraints and the objective function is implicit in all algorithms for NP. In unconstrained and linearly constrained optimization, the 'natural' merit function is simply the objective function. The lack of a natural merit function for nonlinearly constrained problems is part of the reason for the difficulty of solving such problems, and for the great variety of methods. In an SQP method, it should ideally be possible to achieve a sufficient decrease in the merit function at each iteration, and the need to reduce the merit function should not restrict the rate of convergence of the SQP method. However, no known merit function is perfect.

Currently the most popular merit function is the l_1 penalty function P_1 (3); its use as a merit function was popularized by Han [25]. In order to use P_1 as a merit function, the line search must be able to cope with derivative discontinuities. A disadvantage of P_1 as a merit function is the possible inhibition of superlinear convergence [28]. Several strategies have been suggested to overcome this difficulty (e.g., that of Chamberlain et al., [7]) but the l_1 merit function appears to be waning in popularity.

The augmented Lagrangian function (7) has also been used as a merit function [45,46,22]. Further suggestions and references are given in [2], [5], [16], [19] and [48].

10. Issues in SQP methods

In this section, we mention two aspects of SQP methods that may be of interest to the practitioner. First, there may be no feasible point with respect to the constraint linearization (13b), even though feasible points exist with respect to the nonlinear constraints. (In other instances, the linearized constraints are consistent, yet no feasible point exists for the nonlinear constraints.)

In general, infeasible subproblems have been regarded as non-fatal. One way to proceed is to define a damped right-hand side for the linearized version of a violated constraint in (13b) [39]. As the damping factor decreases to zero, the likelihood increases that a feasible point will exist. (A variation of this idea has been implemented by Schittkowski [46].) Several other strategies can be viewed as a more general form of constraint relaxation, in which the idea is to minimize some measure of the violations of the linearized constraints (13b), or a weighted combination of the linearized constraint violations and the quadratic objective function. For example, the 'least-infeasible' point of the original subproblem can be computed, and the linearized constraints (temporarily) relaxed so that the subproblem is guaranteed to have a feasible solution [22]. Other methods of this type have been proposed by Fletcher [18] and Tone [49].

The second point concerns the role of the QP subproblem within an SQP method. Early SQP methods simply used available 'black box' QP codes to solve (13). Assuming that the constraints (13b) are consistent, any reliable method should produce the same solution. The drawback with this approach was entirely practical: the QP codes were unable to save work by taking advantage of information from previous subproblems. When a SQP method is applied to NP, the set of linear constraints active at the solution of the subproblem (13) will eventually become fixed at the correct active set of the nonlinear problem. Therefore, if the QP method could begin with the correct active set, only one QP iteration would be needed to reach optimality. For this reason, much recent research on QP methods has been directed to the development of techniques that enhance efficiency when solving a sequence of related QP problems. In the SQP code of Gill et al. [22], a 'hot start' option has been included that allows information about the active set and portions of the factorizations to be communicated between subproblems. Other issues remain to be resolved in SQP methods — for example, poor scaling and ill-conditioning in the Hessian approximation. Interest has continued in SQP methods in which only a projection of the Hessian needs to be stored [9,34].

11. Methods for large-scale optimization

The practitioner with a large-scale nonlinearly constrained problem faces not only the usual difficulties associated with constraint nonlinearities, but also the immense complexities of storing and factorizing sparse matrices. Fortunately, a general-purpose projected Lagrangian method is available for large-scale problems—the code MINOS [32,33]. MINOS is an implementation of a general projected Lagrangian method, and solves a sequence of linearly constrained subproblems defined by (11) and (12). Each subproblem is solved by a feasible-point active-set method designed for sparse linear constraints [31], retaining a dense quasi-Newton approximation to the projected Hessian. MINOS has successfully solved problems with over 4000 nonlinear constraints and 8000 variables. The method in MINOS is most effective when the number of constraints active at the solution is relatively large, since the number of degrees of freedom determines the size of the projected Hessian approximation that must be stored. (On typical mainframes, the number of degrees of freedom should not be more than three or four hundred. Fortunately, this is not a limitation in many practical applications.)

Despite the success of MINOS, it is clearly desirable to develop alternatives. An SQP method would seem desirable for certain categories of large-scale problems—for example, those in which function and constraint evaluations are expensive, or the number of degrees of freedom is very

large (i.e., relatively few constraints are active at the solution). Unfortunately, no general-purpose version of an SQP method for large-scale problems has yet been produced. However, since an SQP code can be constructed from general-purpose software for large-scale linear equations, it has been possible to develop an implementation for specific problem categories, such as that described in the next section. See also [21].

Another possibility is to use MINOS to solve the QP subproblems (see Section 12 for an example). Such a method would tend to be effective on the same problem type as MINOS, i.e., where the number of degrees of freedom is less than a few hundred. However, its speed of execution might be faster than that of MINOS, since the number of calls to user-supplied subroutines would typically be smaller.

In order to be effective on large-scale problems, an SQP method should be able to exploit sparsity in the Hessian of the Lagrangian function. (MINOS exploits sparsity only in the constraints.) At first, it might appear unlikely that the Hessian of the Lagrangian function would be sparse, since it is a weighted sum of the Hessians of the objective function and the constraints. However, sparsity in the gradient of a nonlinear constraint always implies sparsity in its Hessian matrix. (If a constraint gradient contains at most five nonzero components, the corresponding Hessian matrix can have at most 25 nonzero elements.) Furthermore, in practice there is often considerable overlap in the positions for nonzero elements in the Hessians of different constraints. Thus, the Hessian of the Lagrangian function may be comparable in sparsity to the Jacobian of the constraints.

In most SQP methods that have been used for dense problems, the matrix in (13a) is a positive-definite quasi-Newton approximation to the Hessian of the Lagrangian (or augmented Lagrangian) function. Unfortunately, the success of quasi-Newton methods on dense problems has not been realized when the approximate Hessian is required to be sparse — even in the unconstrained case (see, e.g., [47]). Fortunately, in many large-scale applications it is possible to obtain analytic second derivatives of the problem functions. Therefore, SQP methods seem especially promising for large problems in which the Hessian matrices are known or are approximated by sparse finite-differencing techniques (see, e.g., [12], [41] and [10]).

12. A case history: the optimal power flow problem

To illustrate the advanced state of optimization algorithms, we briefly describe the development of an algorithm to solve the optimal power flow (OPF) problem. In broad terms, an OPF problem addresses the issue of how best to generate and distribute electrical power. The details of formulation of OPF problems vary widely in both form and intent. For example, the objective may range from online operating control of an electrical power system to long-range planning for the same system. In the latter case, the primary issues may be investment strategies or the utilization of existing equipment. All OPF problems are closely related mathematically. However, the different needs of the problem may affect the solution method. For example, an online problem clearly needs to be solved in 'real time'. Furthermore, since the computed solution will be implemented automatically, it is vital that any algorithm used in this context be robust.

An OPF problem is of the following form:

$$\text{OPF:} \quad \underset{x \in \mathbb{R}^n}{\text{minimize}} F(x) \text{ subject to } c(x) = 0, \quad l \leqslant x \leqslant u.$$

The function $F(x)$ is almost always very simple, and is sometimes piecewise linear. Note the unusual feature that there are no general linear constraints. (Large-scale problems usually have many more general linear constraints than nonlinear constraints.) In a typical large OPF problem, $n \approx 5000$ and $m \approx 4500$. However, if an operator is concerned about the system being secure in the event of an unexpected contingency (e.g., a transmission line being struck by lightning), the problem may be an order of magnitude larger.

The nonlinear constraints arise mainly from modelling the distribution network. For example, Kirchoff's law holds at each node in the network. Since each constraint involves only a few variables, the Jacobian matrix of $c(x)$ is sparse. Moreover, the functions consist primarily of products of the variables and simple trigonometric functions, and hence both first and second derivatives can easily be obtained.

The OPF problem was first attempted in the early 1960s, when good algorithms to obtain a feasible point to the power flow equations (the constraints) were discovered. However, even solving this problem proved difficult in some cases. Progress was inhibited for a number of reasons. It was generally believed among power engineers that an OPF algorithm should generate only feasible iterates (or allow at most mild infeasibilities), based on the premise that a useable solution would then be available if the algorithm terminated prematurely. Although such a feature is desirable in an unreliable algorithm, it is of negligible significance to a highly robust algorithm—especially when insisting on such a property may actually degrade reliability.

Most of the original algorithms developed for the OPF problem were first-order techniques, and hence little better than projected steepest descent. Furthermore, use of these methods led to an overly pessimistic picture of the degree of difficulty of the OPF problem. In particular, engineers speculated that there were many local optima, and that the objective function in the feasible space was "rippled". These false conclusions arose because the algorithms terminated at many different points, depending on the starting point. (An unreliable algorithm frequently exhibitits such behavior.)

In our early attempts to solve the OPF problem, we used MINOS directly on the general problem. Its performance indicated that, despite the 'folklore', the OPF problem was reasonably well behaved. One of the benefits of off-the-shelf routines is that they enable an approach to be tried without too much effort. Although more efficient algorithms may be devised later, the ability to solve the problem—even relatively inefficiently—gives much vital information.

A direction for improvement was suggested because MINOS does not utilize second derivatives, which are available for the OPF problem. It is a 'rule of thumb' in optimization that the higher the level of user-supplied derivatives, the more efficient and reliable the algorithm, where 'efficiency' is measured by the number of iterations needed to obtain a good approximation to the solution. However, for large-scale problems the time required to generate the iterates may well dominate the total computational effort. The usefulness of explicit second-derivative informations for large-scale problems depends on the sparsity of the Hessian of the Lagrangian function (see Section 11).

In the OPF problem, the sparsity of this Hessian was similar to that of the Jacobian. The obvious method to try, given its success on small dense problems, was an SQP algorithm. However, an SQP iteration requires the solution of a large-scale QP, for which no specialized software was available. Once again, the general-purpose code MINOS was applied to solve the QP subproblems, without taking advantage of the quadratic form of the objective function. A major benefit of using MINOS in this context was that QP codes usually expend substantial

effort to compute the initial factorization(s). In contrast, MINOS could use an approximate factorization of the projected Hessian from the previous subproblem, and hence was more efficient in later iterations than a standard QP code.

This approach demonstrated that an SQP approach was remarkably efficient, provided the QP subproblems could also be solved efficiently. It also showed that older methods had rarely if ever obtained an accurate solution. The new method was so reliable that its failure led to the discovery of several long-standing errors in the models. However, one drawback of the SQP approach based on MINOS is that its efficiency depends on having only a relatively small number of degrees of freedom (typically, a few hundred).

Many theoretical and practical difficulties are associated with the development of a large-scale SQP algorithm based on second derivatives. Fortunately, it is somewhat easier to develop a successful method for a specific application, such as the OPF problem. For example, the QP subproblem may have an unbounded solution for a general problem, since the Hessian of the Lagrangian can be indefinite. This cannot happen in the OPF problem because of the presence of bounds on all the variables. Indeed, since the upper and lower bounds tend to be quite close, the presence of indefiniteness is unlikely to lead to inefficiency.

Another difficulty that may arise on general problems is that a QP subproblem may have no feasible solution (see Section 10). For the OPF problem, an infeasible subproblem indicates that no feasible point exists for the original problem. In such circumstances, the original problem can always be altered in a meaningful way (by increasing the number of variables) to ensure a feasible solution.

The OPF example illustrates that new approaches can be successfully applied to a particular problem even when they are not yet fully developed for general problems. The implication is that practitioners can exploit the latest developments in optimization, and need not wait until all issues have been resolved. In fact, certain methods unsuitable for general use may still have a useful role in particular applications.

The OPF problem also highlights the interaction between the development of models and algorithms. Success in solving a given model inevitably leads to formulation of more sophisticated models.

13. Conclusions

Since the 1960s, there has been enormous progress in our ability to apply general-purpose methods to solve nonlinearly constrained optimization problems. The typical user today has a good chance of being able to solve a problem with an off-the-shelf routine.

Much work still remains to analyze how existing SQP and projected Lagrangian methods can be made more reliable. In addition, certain applications will benefit considerably from the adaption of new approaches to cater for special features.

References

[1] J. Abadie and J. Carpentier, Generalization of the Wolfe reduced-gradient method to the case of nonlinear constraints, in: R. Fletcher, Ed., *Optimization* (Academic Press, London/New York, 1969) 37–49.

[2] M.C. Bartholomew-Biggs, Line search procedures for nonlinear programming algorithms with quadratic programming subproblems, Report 116, Numerical Optimisation Centre, Hatfield Polytechnic, Hatfield, England, 1981.
[3] D.P. Bertsekas, *Constrained Optimization and Lagrange Multiplier Methods* (Academic Press, New York/London, 1982).
[4] M.C. Biggs, Constrained minimization using recursive equality quadratic programming, in: F.A. Lootsma, Ed., *Numerical Methods for Non-Linear Optimization* (Academic Press, London/New York, 1972) 411–428.
[5] P.T. Boggs and J.W. Tolle, A family of descent functions for constrained optimization, Report 81-3, Department of Mathematics, University of North Carolina, Chapel Hill, NC, 1981.
[6] C.W. Carroll, The created response surface technique for optimizing nonlinear restrained systems, *Operations Research* **9** (1961) 169–184.
[7] R.W. Chamberlain, C. Lemaréchal, H. Pedersen and M.J.D. Powell, The watchdog technique for forcing convergence in algorithms for constrained optimization, *Math. Prog. Study* **16** (1982) 1–17.
[8] T.F. Coleman, A superlinear penalty function method to solve the nonlinear programming problem, Ph.D. Thesis, University of Waterloo, Ontario, Canada.
[9] T.F. Coleman and A.R. Conn, Nonlinear programming via an exact penalty function, *Math. Prog.* **24** (1982) 123–161.
[10] T.F. Coleman and J.J. Moré, Estimation of sparse Hessian matrices and graph coloring problems, Report 82-535, Department of Computer Science, Cornell University, New York, 1982.
[11] A.R. Conn, Constrained optimization using a non-differentiable penalty function, *SIAM J. Numer. Anal.* **10** (1973) 760–779.
[12] A.R. Curtis, M.J.D. Powell and J.K. Reid, On the estimation of sparse Jacobian matrices, *J. Inst. Maths. Applics.* **13** (1974) 117–119.
[13] G.B. Dantzig, *Linear Programming and Extensions* (Princeton University Press, Princeton, NJ, 1963).
[14] J.E. Dennis and J.J. Moré, Quasi-Newton methods, motivation and theory, *SIAM Review* **21** (1977) 443–469.
[15] J.E. Dennis and R.B. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations* (Prentice-Hall, Inc., Englewood Cliffs, NJ, 1983).
[16] L.C.W. Dixon, Exact penalty functions in nonlinear programming, Report 103, Numerical Optimisation Centre, Hatfield Polytechnic, Hatfield, England, 1979.
[17] A.V. Fiacco and G.P. McCormick, *Nonlinear Programming: Sequential Unconstrained Minimization Techniques* (Wiley, New York/Toronto, 1968).
[18] R. Fletcher, *Practical Methods of Optimization, Volume 2, Constrained Optimization* (Wiley, New York/Toronto, 1981).
[19] R. Fletcher, Penalty functions, in: A. Bachem, M. Grötschel and B. Korte, Eds., *Mathematical Programming: The State of the Art* (Springer, Berlin/Heidelberg/New York/Toronto, 1983) 87–114.
[20] K.R. Frisch, The logarithmic potential method of convex programming, Memorandum of May 13, 1955, University Institute of Economics, Oslo, Norway, 1955.
[21] P.E. Gill, W. Murray, M.A. Saunders and M.H. Wright, Sparse matrix methods in optimization, *SIAM J. Sci. Statist. Comput.* **5** (1984) 562–589.
[22] P.E. Gill, W. Murray, M.A. Saunders and M.H. Wright, User's guide for SOL/NPSOL: a Fortran package for nonlinear programming, Report SOL 84-7, Department of Operations Research, Stanford University, CA, 1984.
[23] P.E. Gill, W. Murray, M.A. Saunders, J.A. Tomlin and M.H. Wright, On projected Newton barrier methods for linear programming and an equivalence to Karmarkar's projective method, Report SOL 85-11, Department of Operations Research, Stanford University, CA, 1985.
[24] P.E. Gill, W. Murray and M.H. Wright, *Practical Optimization* (Academic Press, London/New York, 1981).
[25] S.-P. Han, Superlinearly convergent variable metric algorithms for general nonlinear programming problems, *Math. Prog.* **11** (1976) 263–282.
[26] S.-P. Han, Dual variable metric algorithms for constrained optimization, *SIAM J. Control and Optimization* **15** (1977) 546–565.
[27] M.R. Hestenes, Multiplier and gradient methods, *J. Opt. Theory* Applics. **4** (1969) 303–320.
[28] N. Maratos, Exact penalty function algorithms for finite-dimensional and control optimization problems, Ph.D. Thesis, University of London, 1978.
[29] D.Q. Mayne and N. Maratos, A first-order, exact penalty function algorithm for equality constrained optimization problems, *Math. Prog.* **16** (1979) 303–324.

[30] W. Murray and M.H. Wright, Computation of the search direction in constrained optimization algorithms, *Math. Prog. Study* **16** (1982) 63–83.
[31] B.A. Murtagh and M.A. Saunders, Large-scale linearly constrained optimization, *Math. Prog.* **14** (1978) 41–72.
[32] B.A. Murtagh and M.A. Saunders, A projected Lagrangian algorithm and its implementation for sparse nonlinear constraints, *Math. Prog. Study* **16** (1982) 84–118.
[33] B.A. Murtagh and M.A. Saunders, MINOS User's Guide, Report SOL 83-20, Department of Operations Research, Stanford University, CA, 1983.
[34] J. Nocedal and M. Overton, Projected Hessian updating algorithms for nonlinearly constrained optimization, Report 95, Department of Computer Science, Courant Institute of Mathematical Sciences, New York University, New York, 1983.
[35] J.M. Ortega and W.C. Rheinboldt, *Iterative Solution of Nonlinear Equations in Several Variable* (Academic Press, London/New York, 1970).
[36] T. Pietrzykowski, An exact potential method for constrained maxima, *SIAM J. Numer. Anal.* **6** (1969) 299–304.
[37] M.J.D. Powell, A method for nonlinear constraints in minimization problems, in: R. Fletcher, Ed., *Optimization* (Academic Press, London/New York, 1969) 283–298.
[38] M.J.D. Powell, Introduction to constrained optimization, in: P.E. Gill and W. Murray, Eds., in *Numerical Methods for Constrained Optimization* (Academic Press, London/New York, 1974) 1–28.
[39] M.J.D. Powell, A fast algorithm for nonlinearly constrained optimization calculations, Report DAMTP 77/NA 2, University of Cambridge, England, 1977.
[40] M.J.D. Powell, Variable metric methods for constrained optimization, in A. Bachen, M. Grötschel and B. Korte, Eds., *Mathematical Programming: The State of the Art* (Springer, Berlin/Heidelberg/New York/Tokyo, 1983) 288–311.
[41] M.J.D. Powell and P.L. Toint, On the estimation of sparse Hessian matrices, *SIAM J. Numer. Anal.* **16** (1979) 1060–1074.
[42] S.M. Robinson, A quadratically convergent algorithm for general nonlinear programming problems, *Math. Prog.* **3** (1972) 145–156.
[43] J.B. Rosen and J. Kreuser, A gradient projection algorithm for nonlinear constraints, in: F.A. Lootsma, Ed., *Numerical Methods for Non-Linear Optimization* (Academic Press, London/New York, 1972) 297–300.
[44] K. Schittkowski, The nonlinear programming method of Wilson, Han, and Powell with an augmented Lagrangian type line search function, *Numer. Math.* **38** (1981) 83–114.
[45] K. Schittkowski, The nonlinear programming method of Wilson, Han, and Powell with an augmented Lagrangian type line search function, *Numer. Math.* **38** (1981) 83–114.
[46] K. Schittkowski, On the convergence of a sequential quadratic programming method with an augmented Lagrangian line search function, Report SOL 82-4, Department of Operations Research, Stanford University, 1982.
[47] D.C. Sorensen, Collinear scaling and sequential estimation in sparse optimization algorithms, *Math. Prog. Study* **18** (1982) 135–159.
[48] R.A. Tapia, Diagonalized multiplier methods and quasi-Newton methods for constrained optimization, *J. Opt. Theory Applics.* **26** (1977) 135–194.
[49] K. Tone, Revisions of constraint approximations in the successive QP method for nonlinear programming, *Math. Prog.* 26 (1983) 144–152.
[50] R.B. Wilson, A simplical algorithm for concave programming, Ph.D. Thesis, Harvard University, 1963.